

講習会

Arduino

8回目

関数を作る

目的

- 数値以外に同じ動作をするプログラムはMain内にいちいち書かずに関数化させることで簡単に扱うことができるようになる。
- マイコンカーのモーターの速度を変えるプログラムはswitch case内のあちこちに出現するけど、毎回全文を書かずに別の場所にプログラムを書いて関数化させることで1行で表すことができる。
- サブルーチンとも呼ばれる。

関数紹介

- void

void

- voidは、関数定義と関数宣言で利用され、その関数が呼び出し元に値を返さないことを意味する。

例

- `void setup(){`
- `// ...`
- `}`
-
- `void loop(){`
- `// ...`
- `}`

intとの違い

- 宣言した文字に数値を返さない。（変数の場合は次回紹介）
- センサーの値を処理して処理用の数値にするというよりはモータの制御などに使われる。
- 数値を返さないため
- `a = b();`のようにはしない。
(aは変数、bは関数としている)

引数

- 数学における関数やコンピュータプログラムにおける手続きにおいて、その外部と値をやりとりするための特別な変数、あるいはその変数の値のことである。

引数なしの場合

- #define LED2 11
- void setup() {
- pinMode(LED2,OUTPUT);
- }
- void LEDset2(){
- digitalWrite(LED2,HIGH);
- delay(500);
- digitalWrite(LED2,LOW);
- delay(500);
- }
- void loop() {
- LEDset2();
- }

- 例ではLED点滅という簡単なプログラムのため直接書いても問題ないけど、複雑化させた時のことを考えるとLEDset2()書くだけで同じ処理をMain内のどこでも行うことができる。
- 非常に楽である。

引数ありの場合

- #define LED1 10
- void setup() {}
- void LEDset(int a){
 - int LED = a;
 - map(LED,0,500,0,255);
 - constrain(LED,0,255);
 - analogWrite(LED1,LED);
 - delay(500);
 - analogWrite(LED1,0);
 - delay(500);
- }
- void loop() {
 - LEDset(500);
- }

- 例ではアナログ出力を使用してLEDの明るさを変えながら点滅させるプログラムになる。
- LEDset()のカッコ内に0から500の数値を入れることで明るさを変えることができる。
- 1か所しか使っていないけど、少し長くなったので複数回使うことを考えるとあらかじめ作っておいた方が楽になる。
- 数値以外は同じ動作をする。

- 次の例ではLEDset3()のカッコ内の数値を入れることでLEDの点滅スピードが変化する。

- #define LED3 12
- void setup() {
- pinMode(LED3,OUTPUT);
- }
- void LEDset3(int b){
- int Speed = b;
- digitalWrite(LED3,HIGH);
- delay(Speed);
- digitalWrite(LED3,LOW);
- delay(Speed);
- }
- void loop() {
- LEDset3(100);
- }

- 引数はいくつでも作ることができる。
 - (int a,int b,int c . . .)
-
- 次の例は複数のLEDをそれぞれ点滅のスピードを変えている。
 - LEDset4()の数値を変えると速さも変わる。
 - LEDset4(50,300)など

- #define LED3 12
- #define LED4 A0
- void setup() {;
- pinMode(LED3, OUTPUT);
- pinMode(LED4, OUTPUT);
- }
- void LEDset4(int c, int d) {
- int Speed1 = c;
- int Speed2 = d;
- digitalWrite(LED3, HIGH);
- delay(Speed1);
- digitalWrite(LED3, LOW);
- delay(Speed1);
- digitalWrite(LED4, HIGH);
- delay(Speed2);
- digitalWrite(LED4, LOW);
- delay(Speed2);
- }
- void loop() {
- LEDset4(50, 300);
- }

- 作った関数をMain内に書くことで動作するようになる。
- Main内に書かないと扱えないが、作った関数の中に作った関数を入れることもできる。

終わりに

- 簡単な例をあげたけどこれ以外にもいくらかでも応用を効かせることができる。
- 同じようなプログラムを書かずに済むため容量の節約にもなる。
- 次回は変数を作るを取り扱う。

課題

- voidを使って関数を実際に作ってみる。
- 見本は今回紹介した例を使う。